UNIVERSITAT POLITÈCNICA DE CATALUNYA

Tema 4: Sistemas de Almacenamiento

Departament d'Arquitectura de Computadors

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

Índice

- DMA & Jerarquía
- Discos Duros
- RAIDs
- Discos Ópticos

Transferencia de Información

¿Quién realiza la transferencia de información en una operación de E/S?

- **E/S programada**. La CPU realiza la transferencia.
- Si es el procesador el encargado de mover la información:
  - Memoria → Controlador: out(RDATOS, c)
  - Memoria ← Controlador: c = in(RDATOS)
- Esta situación es perfectamente válida cuando trabajamos con dispositivos que funcionan dato a dato o a frecuencias muy bajas (teclado, mouse, …).
- Sin embargo, **¿qué ocurre cuando hemos de transferir bloques de datos a gran velocidad?**

Controlador de Acceso Directo a Memoria (DMA)

- Por ejemplo, lectura de un bloque de disco:
  - El procesador programaría el controlador.
  - El controlador de disco provocaría una interrupción cada vez que tiene un nuevo dato.

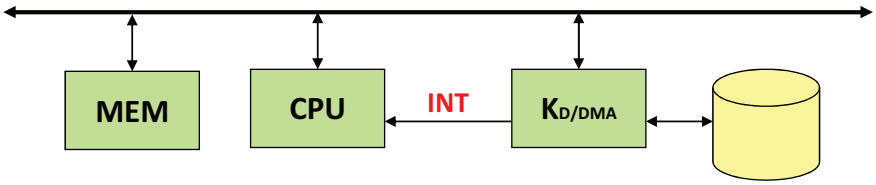
```
void interrupt disco() {
  c = in(RDATOS);
  Memoria[cont]=c;
  cont++;
  if (cont == TAM)
    "acabar";
}
```

Esta RAI se ejecutaría tantas veces como datos fuera a leer. Si el dispositivo es muy rápido, los datos estarán disponibles muy rápidamente, y la CPU será interrumpida constantemente. ¡Apenas podrá hacer otra cosa!

- Hemos encontrado una secuencia de instrucciones que se ha de ejecutar muchas veces y que hace perder mucho tiempo a la CPU.
- La solución obvia es construir un circuito especializado que descarga a la CPU de realizar un trabajo simple, repetitivo y frecuente.

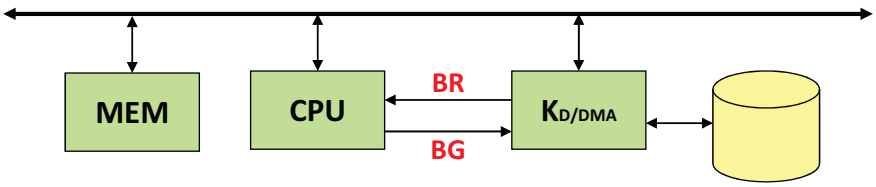
⇒ **CONTROLADOR de ACCESO DIRECTO a MEMORIA (DMA)**

Transferencia vía DMA



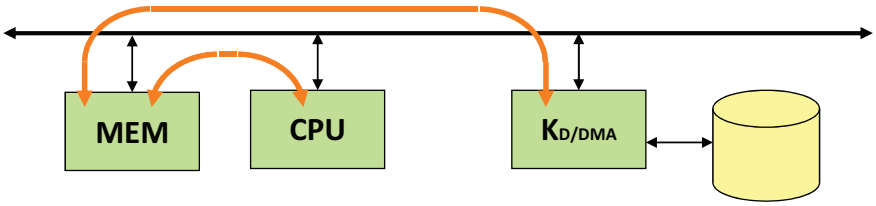
- Cuando el procesador quiere hacer una operación con el disco, ha de programar el disco y el DMA. **Programar el DMA** es muy simple: indicar si es una **lectura o una escritura**, **cuántos datos** se transfieren y **dónde** dejarlos (o de dónde leerlos).
- A partir de aquí, la CPU puede hacer cualquier otra cosa.
- Cada vez que el disco tiene un nuevo dato (o bloque de datos), el DMA se encarga de escribirlo (o leerlo) en memoria en la posición adecuada.
- Cuando se han transferido todos los datos, la CPU se ha de **sincronizar** con el disco. Normalmente, el disco (o DMA) genera una **interrupción**.
- La transferencia por DMA sólo tiene sentido cuando en la transferencia está involucrado un **bloque de datos**.

Transferencia vía DMA



- **El procesador y el KDMA se han de coordinar para acceder a Memoria sin conflictos:**
  - Protocolo por **robo de ciclo**:
    - ✓ El procesador tiene prioridad en el acceso al bus de Memoria.
    - ✓ Cuando el KDMA necesita acceder a Memoria, activa BR.
    - ✓ Si el procesador no necesita acceder a Memoria, contesta activando BG.
    - ✓ El KDMA realiza la transferencia (1 dato).
    - ✓ Para acabar, el KDMA desactiva BR y el procesador desactiva BG.
    - ✓ El procesador tiene nuevamente acceso al bus de memoria.
  - Protocolo por **transferencia a ráfaga**:
    - ✓ El protocolo es similar, pero ahora en vez de transferir datos individuales, se transfieren múltiples datos antes de desactivar BG.

DMA y Jerarquía de Memoria

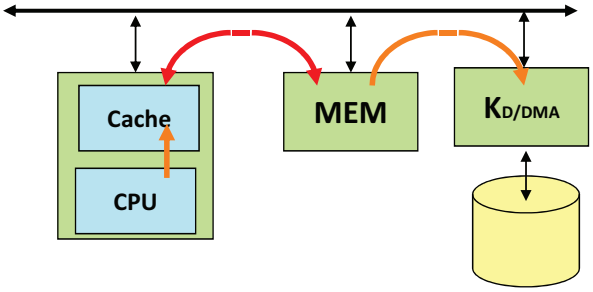


- En un computador, tanto el **procesador como el K_{DMA} pueden acceder a Memoria**.
- Esta situación puede provocar:
  - Problemas de **COHERENCIA** con la Memoria Cache
  - Problemas con la **TRADUCCIÓN de DIRECCIONES**

DMA y Jerarquía de Memoria

Problema de COHERENCIA 1

1. La **CPU escribe** un dato X en la cache.
2. Una vez escrito, el dato sigue permaneciendo en cache hasta que la línea sea substituida.
3. Se programa una **escritura en disco** del bloque de memoria donde estaba X (X tiene un valor distinto en Memoria Principal que en Memoria Cache con lo que escribimos un **VALOR INCORRECTO**).

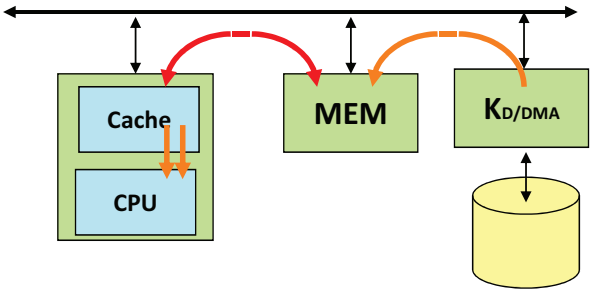


Existen diversas soluciones para este problema:
- Usar una cache **Write Trough**
- Que el K_{DMA} sólo pueda acceder a zonas de MP **NO-CACHEABLES**
- Vaciar la cache cada vez que se lanza una operación con el K_{DMA}

DMA y Jerarquía de Memoria

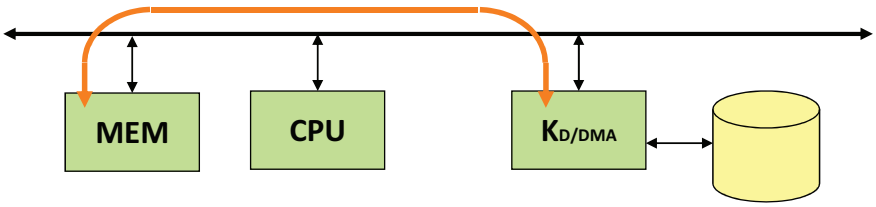
Problema de COHERENCIA 2

1. La **CPU lee** un dato X y lo trae a la cache.
2. Una vez leído el dato, el dato sigue permaneciendo en cache hasta que la línea sea substituida.
3. Se programa una **lectura de disco** y se lee un bloque de datos que incluye la posición donde estaba X (X toma un nuevo valor en Memoria Principal).
4. La **CPU lee** X de nuevo, pero obtiene el valor almacenado en la cache y **NO el VALOR CORRECTO** que está en Memoria Principal.
   - Que la cache sea write through **NO soluciona** el problema.



Existen diversas soluciones para este problema:
- Que el K_{DMA} sólo pueda acceder a zonas de MP NO-CACHEABLES
- Vaciar la cache cada vez que se lanza una operación con el K_{DMA}

DMA y Jerarquía de Memoria



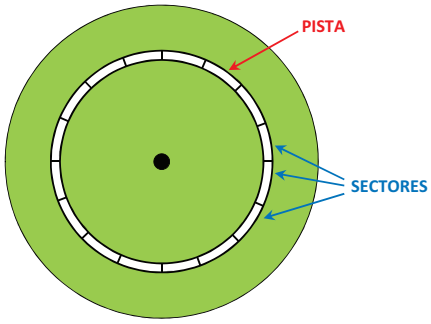
Problemas con la TRADUCCIÓN de DIRECCIONES.

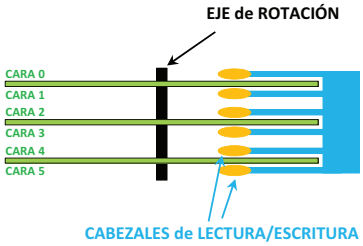
- El **K_{DMA} ha de acceder a Memoria Principal** utilizando **direcciones físicas**.
- Eso implica que cuando el **SO** quiera hacer una operación con el K_{DMA}, tendrá que acceder a la **tabla de páginas** y programarlo con **direcciones físicas**.

Disco Duro

Dispositivo de Almacenamiento Masivo

- Gran Capacidad
- Memoria No volátil
- Tecnología Magnética
- Elemento Mecánico
- Organizado en:
  - Caras
  - Pistas (Cilindros)
  - Sectores
- Métodos de Acceso
  - **Modo CHS** (Cylinder Head Sector). Se accede a disco con especificaciones geométricas: Cilindro, Cara y Sector.
  - **Modo LBA** (Logical Block Addressing). Los sectores están numerados de 0 a N-1 (N sectores lógicos por disco). Se accede a disco utilizando 1 número de sector lógico. Requiere un mecanismo de traducción interno.





Disco Duro

Parámetros para evaluar el rendimiento de un disco duro:

- **Average Seek Time**: Coste en media de situar el cabezal en el cilindro al que se quiere acceder. Se mide en **milisegundos**.
- **Latency**: Coste en media de situar el cabezal sobre el sector al que se quiere acceder. Se mide en **milisegundos**. Depende de la velocidad de giro del disco duro (R.P.M.).
- **R.P.M.**: **Revoluciones por minuto**. En modo «normal» el disco está siempre girando. Valores típicos: 7200 (PC), 4200 (portátil), 10000 y 15000 (servidor).
- **Average access time**: Average Seek Time + Latency
- **Transfer rate**: Velocidad de transferencia, medida en **MB/s**. Velocidad típica: 100-300 MB/s.
- **Cache**: Memoria secundaria en donde se almacena temporalmente la información leída de disco (o pendiente de escribir, equivale a un buffer de escritura). Posibles estrategias de prefetch. Valores típicos: 8-64 **MB**.
- **MTTF**: Tiempo medio entre fallos. Valores típicos: **1 000 000 – 1 600 000 horas**.
- **Interfaz**: Forma de comunicación entre el Disco Duro y el Computador. Ejemplos actuales: **SATA**, **SCSI**, …

Disco Duro

- Los fabricantes miden la fiabilidad de un disco duro con el MTTF.
- Un MTTF de 1 200 000 horas nos indica que el tiempo de vida medio de un disco duro está en **140 años (¡!)**.
- **Sin embargo, la vida útil de un disco duro está en 5-6 años (unas 50.000 horas).**
- Esta discrepancia tiene que ver con la forma en que se calcula este valor.
- Una medida más útil, que se calcula a partir del MTTF, es el porcentaje de discos que fallan en 1 año: AFR (Annual Failure Rate):

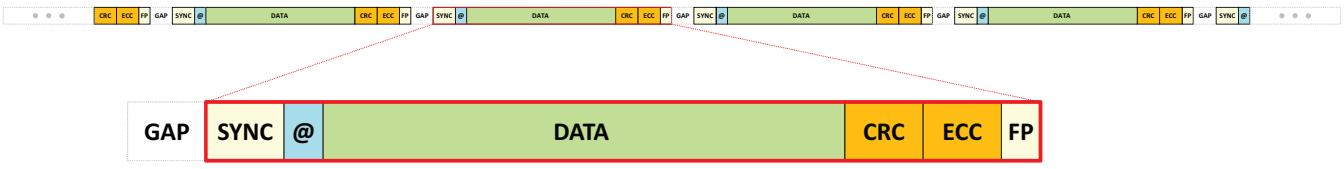
$$\text{Discos que fallan en un año} = \frac{1000 \text{ discos} \times 8760 \text{ horas/disco}}{1200000 \text{ horas/fallo}} = 7{,}3 \longrightarrow \textbf{El AFR es } \mathbf{0{,}73\%}$$

- Sin embargo, trabajos recientes muestran que el AFR real está en **2-4%** (1).
  - MTTF ≈ 200.000 – 400.000 horas
- Este valor aumenta hasta un **8,6%** en discos con 3 años de antigüedad (2).
  - MTTF ≈ 100.000 horas

(1) Bianca Schroeder, Garth Gibson. "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean too you?" 5th Usenix Conference on File and Storage Technologies (FAST 2007).
(2) Eduardo Pinheiro, Wolf-Dietrich Weber and Luiz André Barroso. " Failure Trends in a Large Disk Drive Population " 5th Usenix Conference on File and Storage Technologies (FAST 2007).

Estructura de un sector

- La unidad mínima de acceso a disco es el **sector**.
- Un sector de disco tiene la siguiente estructura:



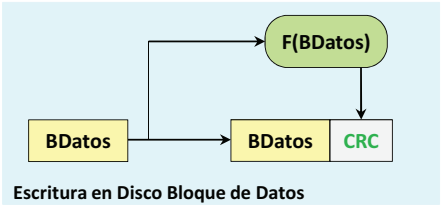
- **GAP**: marca física que indica la separación entre sectores (formateo bajo nivel)
- **SYNC** (Preamble): alrededor de 10 bytes que permite establecer la frecuencia y amplitud con las que se ha grabado la información.
- **@** (Address mark): identificación del sector, más información de estado.
- **DATA**: 512 bytes de datos codificados en RLL
- **CRC**: checksum para comprobar la integridad de los datos (alrededor de 10 bytes).
- **ECC**: información redundante para detectar y corregir errores (alrededor de 40 bytes).
- **FP** (Flush Pad): información interna para sincronización.

El detalle de la estructura depende del fabricante. Desde el punto de vista del programador (Sistema Operativo) leemos o escribimos sectores de 512 bytes. El controlador del disco duro se encarga de que la información se lea o escriba correctamente.

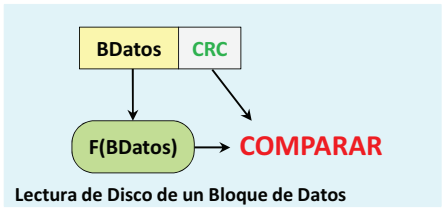
CRC (Cyclic Redundancy Check)

- Algoritmo utilizado para comprobar la integridad de un bloque de información después de una operación de lectura o transmisión.
- Es un algoritmo derivado de la división de polinomios con XOR, en vez de división.
- Un algoritmo (simple) en C podría ser similar a:

```
resto = BloqueDatos;
for (bit=0; bit<M; bit++) {
  resto = resto ^ POLINOMIO;
  resto = (resto << 1);
}
CRC = resto;
```

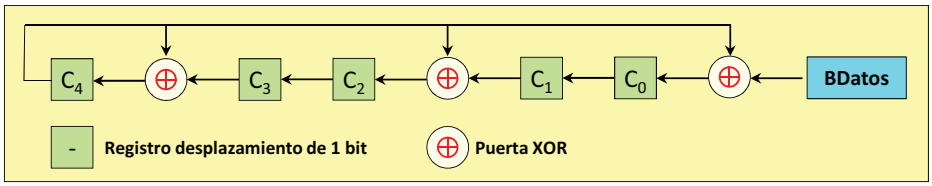


Escritura en Disco Bloque de Datos



Lectura de Disco de un Bloque de Datos

- Algoritmo hardware:



«División» del polinomio $X^5+X^4+X^2+1$

Usado en redes RDSI

ECC (Error Correcting Code)

- Junto con cada sector de 512 bytes almacenado en disco se incluye información redundante para detectar y corregir errores.
- El algoritmo utilizado es el Reed-Solomon (1).
- Utilizado en múltiples entornos: almacenamiento magnético, almacenamiento óptico, modems de alta velocidad, canales de transmisión de datos, …
- Es más fácil de implementar y requiere menos bits redundantes que otros algoritmos para el mismo nivel de eficacia. Es especialmente eficaz para ráfagas de bits erróneos (2).
- El firmware del disco duro se encarga de gestionar la detección y corrección de errores. El usuario (o SO) sólo recibirá el bloque de datos que ha solicitado o una notificación de error
- La eficacia del algoritmo ECC depende del número de bits redundantes que se almacenen.
  - Incluir más bits de ECC implica que el sistema es más robusto, pero significa menos sectores por pista.
  - Un sistema más robusto permite aumentar la densidad de datos.
  - Más bits de ECC implica que el controlador correspondiente ha de ser más potente.
  - Los ingenieros de diseño de discos duros han de tener en cuenta estos factores para decidir cuántos bits ECC incluir para cada sector.

(1) I. S. Reed and G. Solomon. "Polynomial Codes Over Certain Finite Fields". SIAM Journal of Applied Math. , pp. 300-304, vol. 8, 1960.
(2) Bernard Sklar, Reed Solomon Codes , http://ptgmedia.pearsoncmg.com/images/art_sklar7_reed-solomon/elementLinks/art_sklar7_reed-solomon.pdf

S.M.A.R.T.

S.M.A.R.T. (*Self Monitoring Analysis and Reporting Technology*)

- Muchos de los problemas de funcionamiento de un disco duro no ocurren de repente.
- La mayoría son resultado de la degradación (lenta) de los componentes mecánicos o electrónicos.
- La tecnología S.M.A.R.T. monitoriza diferentes parámetros del disco (¡decenas!) con el objetivo de anticipar los problemas, predecir cuando un disco está en situación de riesgo, y avisar al usuario para que lo reemplace.
- La versión actual no sólo predice errores, también corrige algunos.
  - Remapping de sectores erróneos.
  - Los discos vienen con sectores adicionales (no contabilizados)
  - No se detectan «BAD SECTORS» (el disco duro los reasigna).
- Los «avisos de S.M.A.R.T.» están correlacionados con altas probabilidades de fallo:
  - Después del primer aviso, los discos tienen 39 veces más posibilidades de fallar en los siguientes 60 días, que los discos sin avisos (1).
  - Aunque muchos fallos de disco se producen sin aviso previo.

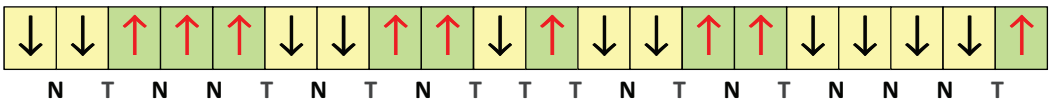
(1) Eduardo Pinheiro, Wolf-Dietrich Weber and Luiz André Barroso. " Failure Trends in a Large Disk Drive Population " 5th Usenix Conference on File and Storage Technologies (FAST 2007).

S.M.A.R.T

- El fabricante decide qué parámetros monitoriza y la importancia de cada uno de ellos.
- Típicamente son una serie de contadores de eventos. Cuando se llega a un cierto umbral (definido por el fabricante) se realiza un aviso.
- Alguno de esos parámetros son:
  - *Read Error Rate*. Porcentaje de errores de lectura de sector.
  - *Throughput Performance*. Tasa de transferencia del disco.
  - *Reallocated Sectors Count* . Número de sectores reasignados
  - *Seek Time Performance*. Tiempo medio de búsqueda
  - *Power-on hours*. Horas de trabajo acumuladas.
  - *Recalibration Retries*. Cuantas veces se ha recalibrado el disco.
  - *Temperature*
  - *Current Pending Sector Count*. Número de sectores inestables (pendientes de reasignación)
  - *Soft ECC Correction*. Número de correcciones hechas vía software ECC.
  - *Command Timeout*. Número de operaciones abortadas debidas a timeout del disco.
  - *G-sense error rate*. Número de errores debidos a golpes externos.
  - *Write error rate*: Número de errores cuando escribe un sector.

Codificación

- En un medio físico, para realizar lecturas y escrituras seguras sólo se pueden codificar un número máximo de transiciones por unidad de medida.



- Conocido el número máximo de transiciones, el objetivo es codificar el máximo número de 1's y 0's
- Primeros algoritmos:

Bit pattern	Codificación	# transiciones por bit
0	TN	1 (50%)
1	TT	2 (50%)
0, detrás de 0	TN	1 (25%)
0, detrás de 1	NN	0 (25%)
1	NT	1 (50%)

FM (Frequency Modulated). Obsoleto.
1,5 transiciones por bit

MFM (Modified Frequency Modulated). Utilizado por los "floppy disk" de doble densidad.
0,75 transiciones por bit

Codificación

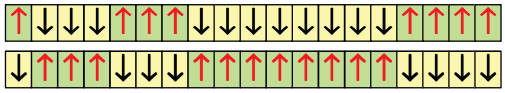
- **Codificación RLL *x,y* (*Run Length Limited*)**
  - x: espacio mínimo entre transiciones
  - y: espacio máximo entre transiciones

Bit pattern	Codificación	# transiciones por bit
11	TNNN	1/2 (25%)
10	NTNN	1/2 (25%)
011	NNTNNN	1/3 (12,5%)
010	TNNTNN	2/3 (12,5%)
000	NNNTNN	1/3 (12,5%)
0010	NNTNNTNN	2/4 (6,25%)
0011	NNNNTNNN	1/4 (6,25%)

Tabla RLL 2,7
0,4635 transiciones por bit

Uso de la codificación RLL:
- Disco duro: RLL 2,7
- CD: variante del RLL 2,10 (EFM, Eigth to Fourteen Modulation)
- DVD: RLL 3,11 (EFM+).
- Blu-Ray: RLL 4,12

- Ejemplo, codificación de la cadena 010110011 en RLL 2,7
  - 010 – 11 – 0011: TNNTNN – TNNN – NNNNTNNN → TNNTNNTNNNNNNNTNNN

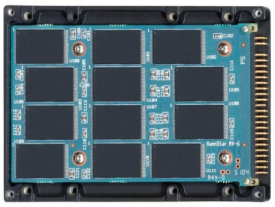


Ejemplos comerciales de discos duros (2008)

Características	Seagate ST22000655SS	Seagate ST31000340NS	Seagate ST973451SS	Seagate ST9160821AS
Diámetro	3,5'' (8,89 cm)	3,5'' (8,89 cm)	2,5'' (6,35 cm)	2,5'' (6,35 cm)
Capacidad (formateado)	147 GB	1000 GB	73 GB	160 GB
Caras	2	4	2	4
Velocidad rotación	15000 RPM	7200 RPM	15000 RPM	5400 RPM
Cache disco interna	16 MB	32 MB	16 MB	8 MB
Interfaz, ancho de banda	SAS, 375 MB/s	SATA, 300 MB/s	SAS, 375 MB/s	SATA, 150 MB/s
Velocidad transferencia	73-125 MB/s	105 MB/s	79-112 MB/s	44 MB/s
Min. seek (read/write)	0,2/0,4 ms	0,8/1 ms	0,2/0,4 ms	1,5/2 ms
Avg. Seek (read/write)	3,5/4 ms	8,5/9,5 ms	2,9/3,3 ms	12,5/13 ms
MTTF (horas)	1400000 @ 25º C	1200000 @ 25º C	1600000 @ 25º C	-
Annual failure rate (AFR)	0,62%	0,73%	0,55%	-
Contact start-stop cycles	-	50000	-	> 600000
Garantía	5 años	5 años	5 años	5 años
Nonrecoverable read errors	<1 sector por 10^{15} bit reads	<1 sector por 10^{15} bit reads	<1 sector por 10^{16} bit reads	<1 sector por 10^{14} bit reads
Temperatura, shock	5-55ºC, 60G	5-55ºC, 63G	5-55ºC, 60G	0-60ºC, 350G
Dimensiones, peso	2,54×10,16×14,73 cm, 681g	2,54×10,16×14,73 cm, 681g	2,03×7,11×9,90 cm, 227 g	1,02×7,11×9,90 cm, 91 g
Consumo (op/idle/standby)	15/11/- W	11/8/1 W	8/5,8/- W	1,9/0,6/0,2
MB/cm³, GB/W	387MB/cm³, 10 GB/W	2,63GB/cm³, 91 GB/W	511MB/cm³, 9 GB/W	2,23GB/cm³, 84 GB/W
Precio en 2008, $/GB	$250, $1,7/GB	$275, $0,3/GB	$350, $5/GB	$100, $0,6/GB

SSD (Solid State Drive)

- Dispositivos de almacenamiento que utilizan memoria no volátil (FLASH) en vez de elementos mecánicos y soporte magnético.
- No es una idea original. Por ejemplo, BATRAM (1986) entre 4 y 20MB de RAM, con una pila recargable para conservar los datos cuando el sistema se apagaba.
- Utilizan NAND Flash
  - Acceso aleatorio a bloques y secuencial dentro de los mismos.
  - Número limitado de escrituras y borrados (entre 10^4 y 10^6).
  - Menor fiabilidad que las NOR Flash (usan ECC para aumentarla)



SSD Ventajas e Inconvenientes

VENTAJAS

- Arranque rápido (no hay elementos mecánicos).
- Gran velocidad de lectura y escritura.
- Baja latencia de lectura y escritura.
- Menor consumo de energía y producción de calor.
- Sin ruido.
- Mejor MTTF que un disco duro.
- Seguridad: borrado seguro e irrecuperable.
- Rendimiento. El tiempo de "búsqueda" es constante.
- El rendimiento no se deteriora mientras el medio se llena. No necesita defragmentación
- Menor peso y tamaño que un disco duro tradicional de similar capacidad.

INCONVENIENTES

- Alto Precio.
- Información no recuperable después de un fallo físico.
- Baja Capacidad.
- Menor tiempo de vida total. Bajo número de ciclos de lectura y escritura.

SSD vs Disco Duro

Atributo	SSD	Disco Duro
Tiempo spin-up	Nulo	Segundos (tiempo para que el disco alcance las RPM)
Tiempo acceso	Alrededor de 0,1 ms	5-10 ms
Latencia lectura	Bajo	Alto
Rendimiento	No depende de la ubicación de los datos	Necesita defragmentación para mantener el rendimiento
Nivel de ruido	Nulo	Apreciable
Fiabilidad Mecánica	Carece de elementos mecánicos	La posibilidad de fallo mecánico aumenta con el tiempo
Factores ambientales	No susceptible a golpes, altitud o vibraciones	Susceptible a golpes, altitud o vibraciones
Magnetismo	No susceptible	Fuentes magnéticas pueden alterar los datos
Peso y tamaño	Ligero y pequeño en comparación con los DD	Más pesados y grandes debido a los elementos mecánicos
Paralelismo	Es posible leer de varios chips a la vez	Varios cabezales, pero todos están en el mismo cilindro
Límite de escrituras	La memoria flash tiene un límite de escrituras	No hay limitaciones.
Coste (Feb 2011)	Entre 1 y 2 euros por GB	Entre 0,05 y 0,1 euros por GB
Capacidad (2010)	Menos de 512GB	Hasta 3 TB
Lect vs Esc	Las escrituras son mucho más lentas	No hay grandes diferencias
Consumo	Entre un 33-50% del consumo de un DD	12-18 W (disco de alto rendimiento), 2W (portátil)

Introducción RAID

RAID (Redundant Array of Inexpensive / Independent Disks)

- Esquema estandarizado de múltiples discos que la industria ha adoptado para almacenar grandes cantidades de datos.
- **Objetivos**
  - Aumentar el rendimiento (ancho de banda)
  - Aumentar la capacidad (muchos discos)
  - Aumentar la fiabilidad (*reliability*) de los datos (tolerancia a fallos) en los sistemas de almacenamiento masivo (Discos Magnéticos)
- **La idea detrás de RAID**: Usar múltiples discos (más capacidad) que operen independientemente y en paralelo para incrementar el ancho de banda y mejorar la fiabilidad.
  - Ficheros distribuidos a través de múltiples discos. El ancho de banda aumenta con el número de discos
  - Se pueden añadir esquemas de redundancia y corrección de errores para mejorar la fiabilidad de los datos.
- **La propuesta original:**
  - David A. Patterson, Garth Gibson and Randy H. Katz. *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. In Proceedings of ACM SIGMOD Conference, pp 109-116, 1988.

Fiabilidad vs rendimiento

- **Fiabilidad**
  - La tecnología RAID protege los datos contra el fallo de una unidad de disco duro. Si se produce un fallo, RAID mantiene el servidor activo y en funcionamiento hasta que se sustituya la unidad defectuosa
  - Los sistemas RAID (excepto RAID 0) suponen la pérdida de parte de la capacidad de almacenamiento de los discos, para conseguir la redundancia o almacenar los datos de paridad
  - Los sistemas RAID profesionales deben incluir los elementos críticos por duplicado: fuentes de alimentación y ventiladores redundantes. De poco sirve disponer de un sistema tolerante al fallo de un disco si después falla por ejemplo una fuente de alimentación que provoca la caída del sistema
- **Rendimiento**
  - La tecnología RAID se utiliza también con mucha frecuencia para mejorar el rendimiento de servidores y estaciones de trabajo (se puede leer o escribir en varios discos simultáneamente)

- Estos dos objetivos, **fiabilidad y mejora del rendimiento**, no se excluyen entre sí
- RAID ofrece varias opciones, llamadas niveles RAID y numeradas desde 0 (RAID 0 – RAID 6). Cada opción proporciona un equilibrio distinto entre tolerancia a fallos, rendimiento y coste

- Un buen punto de consulta es: http://www.storagereview.com

Fiabilidad

- La **fiabilidad** se mide como **tiempo medio entre fallos** (MTTF: *Mean Time To Failure*)
- ¿Qué ocurre en sistemas con múltiples discos?
- La seguridad de un conjunto (*array*) de discos está relacionada con el número de discos
  - MTTF(N discos) = MTTF (1 disco) / N

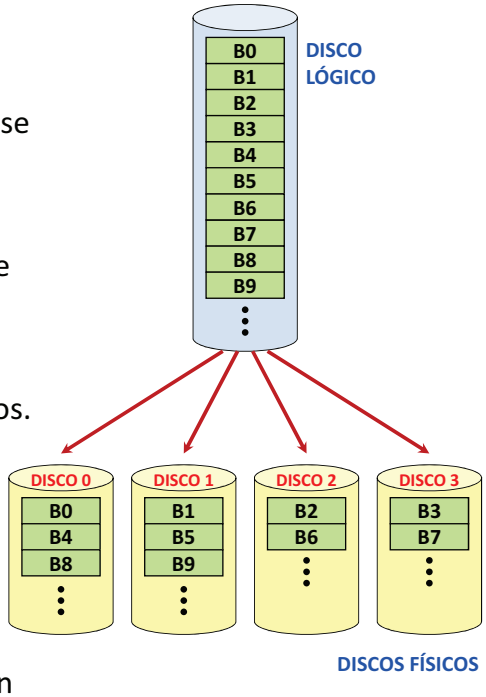
- Ejemplo, MTTF (1 disco) = 1.000.000 horas, con 10 discos:
  - MTTF (10 discos) = 100.000 horas
  - AFR de 1 disco: 0,876%
  - AFR de 10 discos es : 8,76%
- Resultados experimentales, dicen que en discos con 3 años de antigüedad su AFR es 8,6%
  - **¡El AFR de un sistema con 10 discos de 3 años de antigüedad sería 86%!**
- Servidores con más de 10 discos son usuales en el mercado
  - En algunas aplicaciones **no debe** haber problemas durante años
  - Añadir discos a un *RAID* para aumentar capacidad y ancho de banda reduce la fiabilidad
  - **Se necesita redundancia en el RAID para aumentar la fiabilidad**

Introducción: conceptos básicos

- RAID es un **conjunto de unidades físicas** de disco vistas por el sistema operativo como **una única unidad lógica**.
- Los datos se distribuyen de forma entrelazada a través de las unidades físicas. Son posibles distintos niveles de entrelazado:
  - No entrelazado
  - Entrelazado a nivel de tira (*stripe*): cada fichero se divide en bloques llamados tiras que se distribuyen entre los discos. El tamaño típico de las tiras puede ir de 2 a 512 Kbytes
  - Entrelazado a nivel de byte
  - Entrelazado a nivel de bit
- La capacidad de los discos redundantes se usa para almacenar información que garantice la **recuperación de los datos** en caso de fallo del disco.
- Técnicas de redundancia de datos
  - No redundancia
  - *Mirroring*
  - Paridad
  - Códigos *hamming* horizontales
  - Códigos *Reed-Solomon*

RAID 0: Disk Striping

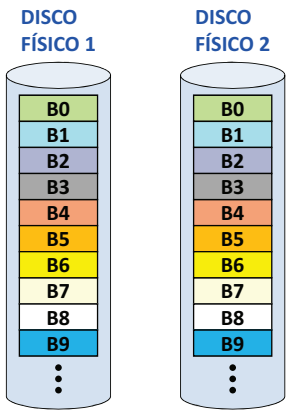
- Distribuye los datos con **entrelazado a nivel de tira** (***stripe***)
- También conocido como "**separación o fraccionamiento (*Striping*)**". Los datos se desglosan en pequeños segmentos y se distribuyen entre los discos del *array*
- Las **unidades de disco conectadas en paralelo** permiten una transferencia simultánea de datos a/de todos ellos, con lo que se obtiene un gran ancho de banda.
- Este nivel de RAID **no ofrece tolerancia al fallo**. Al **no existir redundancia**, RAID 0 no ofrece ninguna protección de los datos. El fallo de cualquier disco del array tiene como resultado la pérdida de los datos y es necesario restaurarlos desde una copia de seguridad
- Este esquema es **aconsejable** en aplicaciones de tratamiento de imágenes, audio o video. En general, **acceso secuencial a ficheros de gran tamaño**. [Siempre y cuando los datos no sean críticos, o sean fácilmente recuperables o generados.]



RAID 1: Mirroring

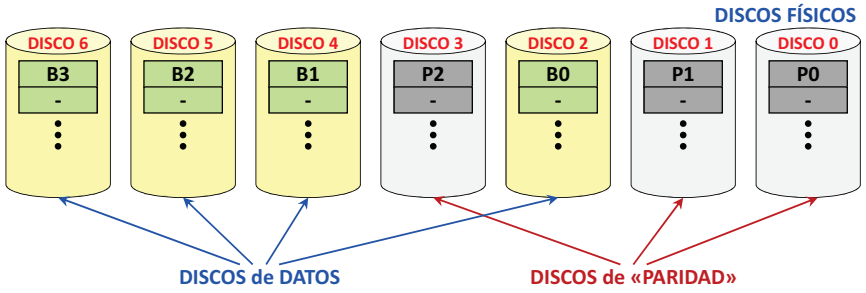
Mirror = ESPEJO

- Utiliza **discos adicionales** sobre los que se realiza una **copia exacta de los datos**
- **Se duplican todos los datos**. De esta manera se asegura la integridad de los datos y la tolerancia al fallo pues, en caso de avería, el RAID sigue trabajando con los discos no dañados sin detener el sistema
- RAID 1 ofrece una excelente **disponibilidad** de los datos mediante la redundancia total de los mismos.
- Los datos se pueden leer desde cualquiera de las copias
- Las escrituras son algo más lentas: se ha de escribir en las dos copias.
- RAID 1 es una alternativa costosa para los grandes sistemas, ya que duplica el coste de los discos



RAID 2: Redundancia a través del código *Hamming*

- Distribuye los datos con **entrelazado a nivel de bit**
- La operación de E/S accede al mismo sector de **todos los discos en paralelo**
- Adapta **la misma técnica ECC** que se usa para detectar y corregir errores en **DRAM**
- El código ECC se intercala a través de varios discos a nivel de bit. El método empleado es el *Hamming*
- Permite detectar y corregir 1 disco que falla o bien detectar que fallan 2 discos.
- **Es un esquema teórico que no se utiliza.**
- El resto de sistemas de corrección (RAID 3 a 6) se basan en tener información de qué disco/s falla/n
- Se dedica un espacio considerable para información redundante.

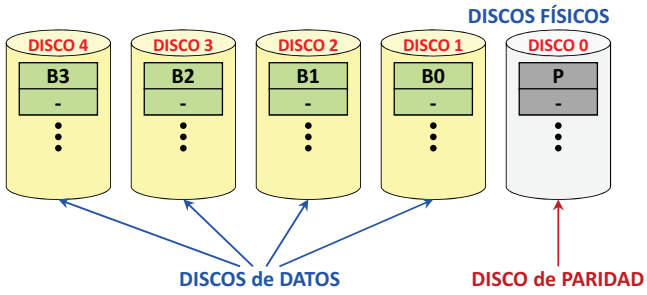


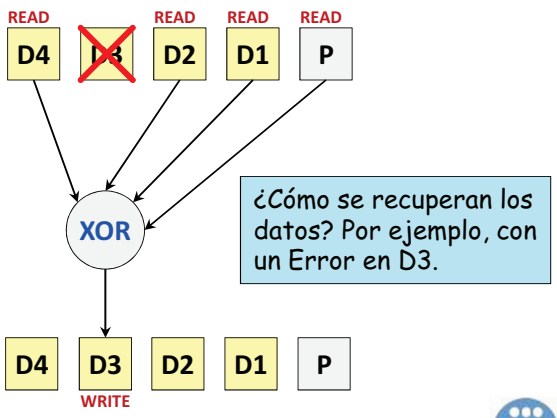
#discos	Discos datos	Discos paridad
7	4	3
15	11	4
31	25	5
63	57	6

Esquema RAID 2 con 7 discos (7,3)

RAID 3: Acceso síncrono con un disco dedicado a paridad

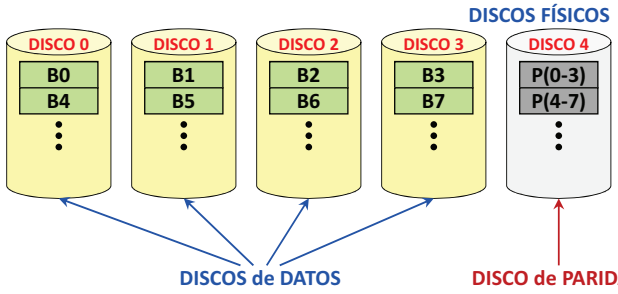
- Distribuye los datos con **entrelazado a nivel de byte (bit en H&P)**
- Dedica **un único disco** al almacenamiento de información de paridad
- La información de ECC del disco (*Error Checking and Correction*) se usa para detectar errores. La recuperación de datos se consigue calculando la OR exclusiva (XOR) de la información registrada en los otros discos
- La operación de E/S accede al mismo sector de **todos los discos en paralelo**
- Su rendimiento de transacción es pobre porque todos los discos del conjunto han de operar conjuntamente





RAID 4: Acceso Independiente con un disco dedicado a paridad

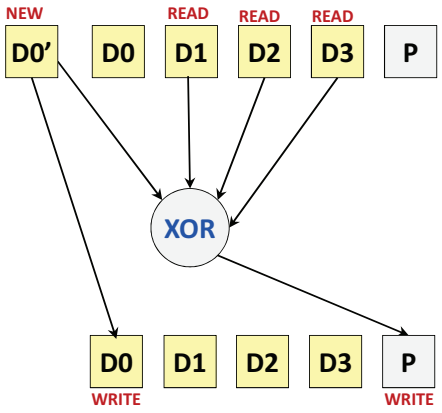
- Es similar a RAID 3 pero con **entrelazado a nivel de tira**
- En RAID 4 **se puede acceder a los discos de forma individual**.
- Basa su tolerancia al fallo en la utilización de un disco dedicado a guardar la información de paridad calculada a partir de los datos guardados en los otros discos.
- **El disco de paridad es el cuello de botella del sistema**
- En caso de avería de cualquiera de las unidades de disco, la información se puede reconstruir en tiempo real mediante la realización de una operación lógica XOR a nivel de tira.
- Sólo se usa **1 disco** para paridad



RAID 4: Acceso Independiente con un disco dedicado a paridad

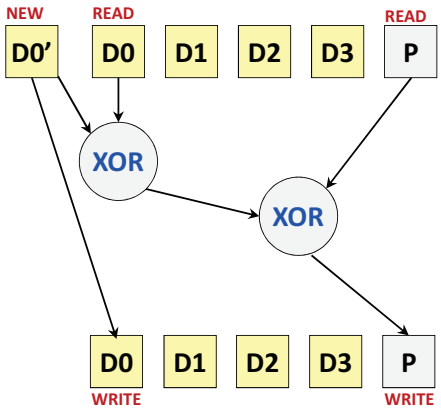
Las escrituras son costosas

- Implementación tipo «RAID 3»



- **Con N discos, necesita N-2 lecturas y 2 escrituras.**

- Implementación en RAID 4

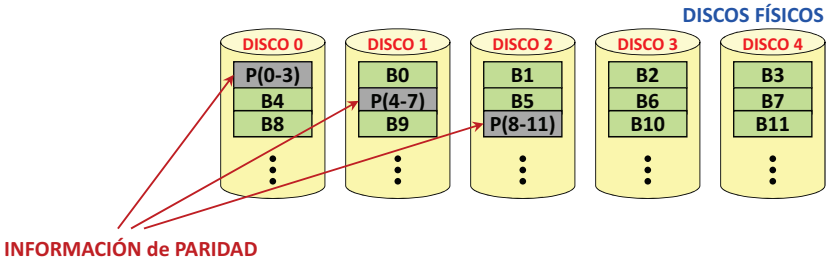


- **Siempre necesita 2 lecturas y 2 escrituras.**

Cuello de botella: SIEMPRE se ESCRIBE en el disco P.

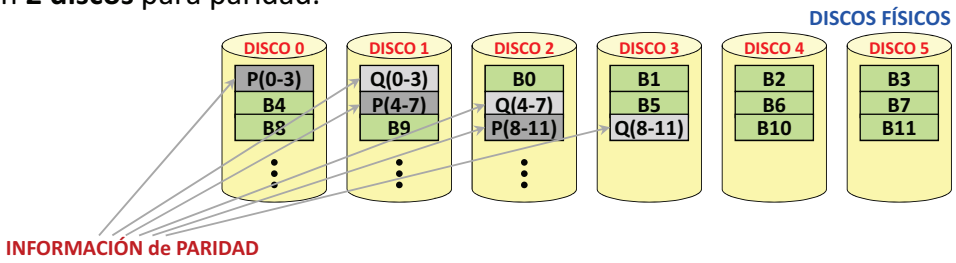
RAID 5: Acceso independiente con paridad distribuida

- Organizado de forma similar al RAID 4. La diferencia es que **los bloques de paridad están distribuidos entre todos los discos**
- Este esquema **evita el cuello de botella que hay en el disco de paridad** de RAID 4.
- RAID 5 no asigna un disco específico a esta misión, sino **un bloque alternativo de cada disco**.
- Distribuir la función de comprobación entre todos los discos disminuye el cuello de botella
- Con una cantidad suficiente de discos puede llegar a eliminarse completamente el cuello de botella, proporcionando una velocidad equivalente a un RAID 0.
- Se puede acceder a los discos de forma **independiente, en paralelo**.
- Una escritura requiere 2 lecturas y 2 escrituras.
- Sólo se usa **1 disco** para paridad.



RAID 6: Acceso independiente con doble paridad

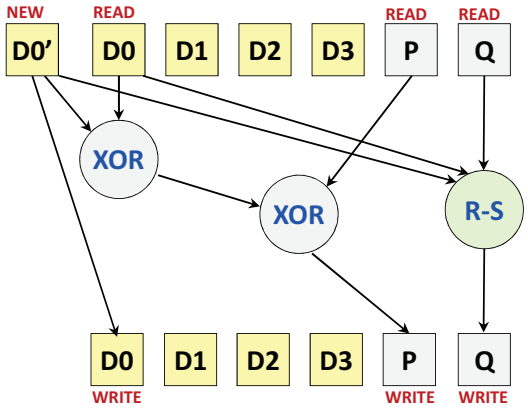
- Similar al RAID 5, pero incluye un **segundo esquema de redundancia distribuido** por los distintos discos. Ofrece tolerancia extremadamente alta a fallos (dos niveles de redundancia)
- Pocos ejemplos comerciales. Su **coste de implementación es mayor** al de otros niveles RAID. Las controladoras que soportan esta doble paridad son más complejas y caras.
- Las dos tiras redundantes son normalmente llamadas P y Q
  - P es la tira de paridad como en RAID 5
  - Q es el segundo nivel de redundancia basado en códigos Reed-Solomon
- Permite recuperar información aunque fallen hasta 2 discos
  - Para conocer mas detalles de cómo se calcula Q y cómo se corrigen 2 errores ver "The mathematics of RAID-6" http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf
- Se usan **2 discos** para paridad.



RAID 6: Acceso independiente con doble paridad

Las escrituras son costosas

- Implementación en RAID 6



- **Siempre necesita 3 lecturas y 3 escrituras.**
- **R-S: Algoritmo Reed-Solomon**
- **P y Q están distribuidos entre todos los discos**

Fiabilidad RAIDs

- **El tiempo entre fallos de 1 disco se aproxima a una distribución exponencial donde:**
  - p = probabilidad de que se produzca un fallo
  - λ = 1/MTTF (failure rate)
  - t = tiempo transcurrido

$$p = 1 - e^{-\lambda t}$$

- **¿Cuál es el tiempo medio entre fallos de un RAID 0 con N discos?**
  - Si falla un solo disco el sistema falla.
  - MTTF(RAID 0) = $MTTF_{disco}$ / N
- **¿Cuál es el tiempo medio entre fallos de otros RAID con N discos?**
  - RAID 3, 4, 5 y 1 (caso particular N=2): Si falla un disco, el sistema sigue operativo.
  - En estos 4 casos, si durante el intervalo MTTR (tiempo de cambiar disco + tiempo de reconstruir la información) falla un segundo disco entonces falla el sistema.
  - RAID 2: no se usa
  - RAID 6: El sistema falla si falla un tercer disco
  - Estudiemos los casos RAID 3,4,5 (N discos) y 1 (N=2)
    - ✓ La expresión para RAID 6 se deja como ejercicio

Fiabilidad RAIDs

- **En un RAID 3,4,5 el sistema falla si falla un segundo disco durante el intervalo MTTR**
  - MTTR = tiempo de cambiar disco + tiempo de reconstruir la información
  - $MTTF_N = MTTF_{disco}/N$ tiempo medio entre fallos para N discos
  - $MTTF_{N-1} = MTTF_{disco}/(N-1)$ tiempo medio entre fallos para N-1 discos
  - Probabilidad de que falle un segundo disco (de N-1 discos que nos quedan):

$$p(\text{fallo 2º disco}) = 1 - e^{-\lambda t} = 1 - e^{-\frac{MTTR}{MTTF_{N-1}}}$$

  - MTTR son unas pocas horas y MTTF pueden ser millones $\Rightarrow$ (MTTR/MTTF es un valor cercano a 0)

$$\text{si } x \approx 0 \Rightarrow 1 - e^{-x} \approx x \text{ por lo que } p(\text{fallo 2º disco}) \approx \frac{MTTR}{MTTF_{N-1}}$$

  - 1/p es el numero de veces esperado que hay que repetir un proceso (falla un disco) con probabilidad p hasta que hay éxito (falla un 2º disco en este caso)
  - Cada vez que falla un disco (de N discos) han transcurrido en media $MTTF_N$ horas, por tanto:

$$MTTF_{RAID} = \frac{MTTF_N \times MTTF_{N-1}}{MTTR} = \frac{MTTF_{disco}^2}{N \times (N-1) \times MTTR}$$

Fiabilidad RAIDs

- **Ejemplos:**
  - $MTTF_{disco}$ = 100.000 horas
  - MTTR = 10 horas
  - N = 10 discos
- **RAID 0**
  - $MTTF_{RAID\ 0}$ = 100.000/10 = 10.000 horas
- **RAID 5**
  - $MTTF_{RAID\ 5}$ = 100.000^2 / (10x9x10) = 11.111.111 horas
- **RAID 1 (N = 2 discos)**
  - $MTTF_{RAID\ 2}$ = 100.000^2 / (2x1x10) = 500.000.000 horas
  - ¡Atención!: no es comparable al resto, sólo usamos 2 discos
- **RAID 6**
  - $MTTF_{RAID\ 6}$ = 13.888.888.889 horas
  - Ejercicio: deducir la expresión para RAID 6

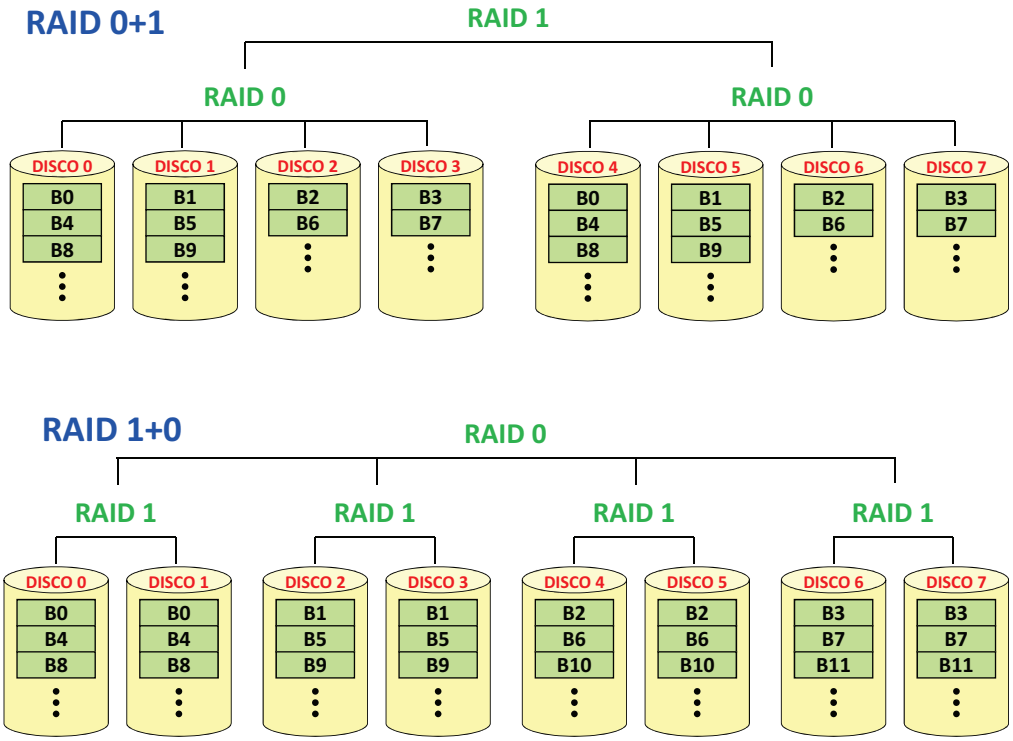
Niveles RAID

Categoría	Nivel RAID	Pros	Contras	Productos comerciales
Acceso Independiente	**RAID 0**. No redundante, striped	Sin overhead	Sin protección	Ampliamente utilizado
Estructura en espejo (mirror)	**RAID 1**. Mirrored	No necesita paridad; recuperación rápida; lecturas rápidas; escrituras más rápidas que otros RAIDs	Mayor overhead	EMC, HP (Tandem), IBM
Acceso Paralelo	**RAID 2**. Estilo memoria ECC	No necesita que falle un disco para diagnosticarse	$\log_2$ N overhead	No usado
	RAID 3. Entrelazado a nivel de byte (bit), paridad	Bajo overhead; alto ancho de banda para lecturas /escrituras grandes	Sin soporte para lecturas / escrituras pequeñas	Storage Concepts
Acceso Independiente	**RAID 4**. Entrelazado a bloques, paridad	Bajo overhead; Más ancho de banda para lecturas pequeñas	Disco paridad es un cuello de botella para las escrituras	Sistemas en red
	RAID 5. Entrelazado a bloques, paridad distribuida	Bajo overhead; Más ancho de banda para lecturas / escrituras pequeñas	Escrituras pequeñas necesitan 4 accesos a disco	Ampliamente utilizado
	RAID 6. Entrelazado a bloques, paridad dual distribuida	Protección contra el fallo en 2 discos	Escrituras pequeñas necesitan 6 accesos a disco; aumenta el overhead	Sistemas en red

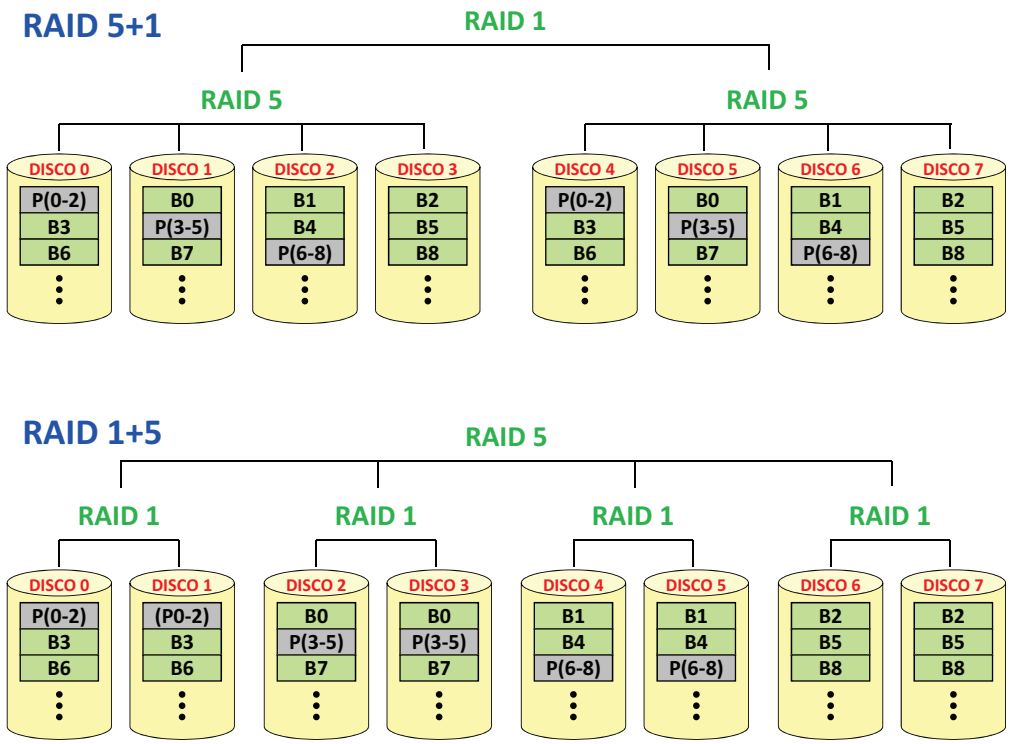
Niveles multi-RAID

- Los distintos niveles de RAID se pueden combinar entre ellos en pares para conseguir las ventajas de ambos niveles de RAID en un único sistema.
- Esquemas multinivel comerciales
  - RAID 0+1 (RAID 01) y RAID 1+0 (RAID 10)
  - RAID 0+5 (RAID 05) y RAID 5+0 (RAID 50)
  - RAID 1+5 (RAID 15) y RAID 5+1 (RAID 51)
- El orden es importante: RAID X+Y ≠ RAID Y+X
  - RAID X+Y consiste en crear grupos de discos con RAID X y después tratar estos grupos como discos individuales para crear un array con RAID Y

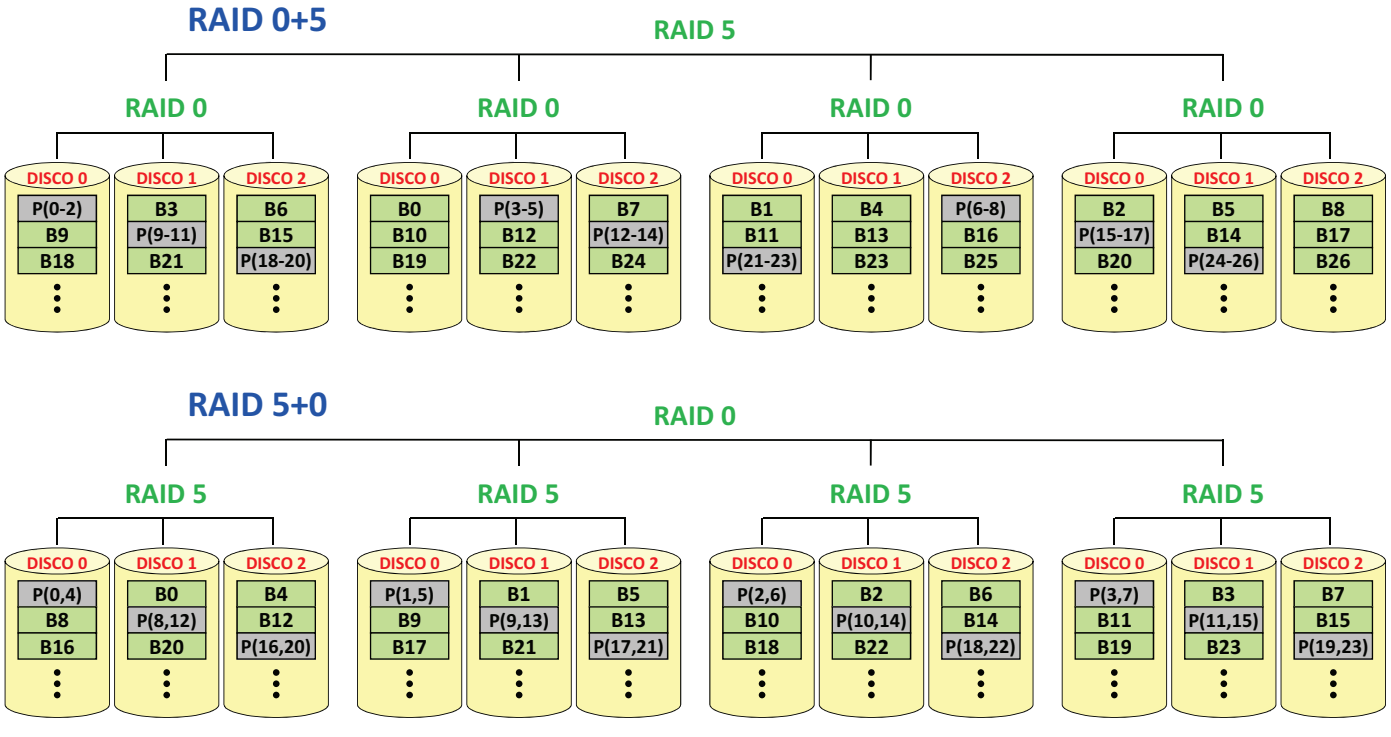
RAID 1+0 vs RAID 0+1



RAID 5+1 vs RAID 1+5



RAID 5+0 vs RAID 0+5



Ejemplo comercial: NEC Storage S2800

Specifications

- Capacity: 54 TB (300GB HDD, RAID5 or RAID6), 240HDD
- Server Interface: 8 Fibre Channel (2Gbps/1Gbps)
- HDD Interface: 2Gbps Fibre Channel
- Support HDD: 36GB (15krpm), 73GB (15k/10krpm), 147GB (15k/10krpm), 300GB (10krpm)
- Support RAID: RAID 1, 5, 6, 10, 50
- Cache Capacity: 16 GB (8GB/Controller, Full Mirror)

Availability

- Full Redundant
- RAID6, Hot Spare Disk
- Advanced error recovery technology "Phoenix"

Functions

- Storage Management: NEC StorageManager
- Replication: DynamicDataReplication, RemoteDataReplication
- Snapshot: DynamicSnapshot
- Performance Monitoring: NEC Storage PerformanceMonitor
- Access Control: AccessControl
- CachePartitioning: CachePartitioning

Discos Ópticos

- La primera generación de discos ópticos (CDs) fue desarrollada por Philips y Sony a mediados de los 80.
  - Una curiosidad: mientras que en los discos magnéticos se habla de “disk”, en los ópticos se habla de “disc”.
- 3 tipos principales: CD, DVD, Bluray
- Características físicas: Mismo tamaño: 120 mm. de diámetro, agujero de 15 mm. No existe el concepto de pista/sector, sino una única pista en espiral. Gira en sentido antihorario (visto desde la cara de datos)
- Son un medio ideal para distribuir software, y material multimedia.
- Los discos ópticos se leen mediante un detector que mide la energía reflejada en la superficie al apuntar a ésta un láser de baja potencia. El láser detecta huecos (pits) (a pesar del nombre son protuberancias) y zonas planas (lands)
- Los pits y lands no representan 0’s y 1’s, se utiliza una codificación **RLL x,y** (p.e. Bluray RLL 4,12)
- Los datos están grabados con **densidad lineal constante**. Aprovechan mejor el medio.
- Para mantener **constante** la **velocidad lineal** del láser el disco ha de girar a **velocidad angular variable**
- Utilizan CRC + **Reed Solomon** para detectar y corregir errores.
- En los CDs de audio no es tan importante la corrección de errores porque luego se han de convertir los datos a señal analógica.